Deadlock and Livelock Exercises

Copyright © James Raynard 2020

Deadlock

- Explain what is meant by deadlock
- Give an example of how deadlock can occur
- Suggest two ways to avoid deadlock

Copyright © James Raynard 2020

Deadlock

- Write a program which causes two threads to deadlock
- Implement your solutions. Verify that the program is no longer affected by deadlock

Copyright © James Raynard 2020

Deadlock avoidance (contd)

- In the following code, why are the unique_lock objects needed when the mutexes are already locked?

```
lock(mutex1, mutex2);
unique_lock<mutex> lk1(mutex1, std::adopt_lock);
unique_lock<mutex> lk2(mutex2, std::adopt_lock);
```

- Rewrite the solution to use unique_lock objects with the defer_lock option

Copyright © James Raynard 2020

Livelock

- What is meant by the term livelock? How does it differ from deadlock?
- Briefly describe a situation where livelock can occur

Copyright © James Raynard 2020

Livelock

- Write a program which demonstrates livelock
- Suggest how livelock could be avoided in your program
- Reimplement your program so it is not affected by livelock

Copyright © James Raynard 2020